WRAPS – A SYSTEM FOR DETERMINING THE PROBABILITY OF PROKARYOTIC PROTEIN ANNOTATION CORRECTNESS

Benjamin K. Nelson
University of Nebraska at Omaha

Follow this and additional works at: https://digitalcommons.unomaha.edu/studentwork

Part of the Computer Sciences Commons

Please take our feedback survey at: https://unomaha.az1.qualtrics.com/jfe/form/SV_8cchtFmpDyGfBLE

Recommended Citation
https://digitalcommons.unomaha.edu/studentwork/2885

This Thesis is brought to you for free and open access by DigitalCommons@UNO. It has been accepted for inclusion in Student Work by an authorized administrator of DigitalCommons@UNO. For more information, please contact unodigitalcommons@unomaha.edu.
WRAPS – A SYSTEM FOR DETERMINING THE PROBABILITY OF
PROKARYOTIC PROTEIN ANNOTATION CORRECTNESS

A Thesis

Presented to the
Department of Computer Science

and the
Faculty of the Graduate College

University of Nebraska

In Partial Fulfillment
Of the Requirements for the Degree

Master of Science

University of Nebraska at Omaha

By

Benjamin K. Nelson

April 2012

Supervisory Committee

Dhundy Bastola, Ph. D

Sanjukta Bhowmick, Ph. D

Zhengxin Chen, Ph. D
ABSTRACT
WRAPS – A SYSTEM FOR DETERMINING THE PROBABILITY OF
PROKARYOTIC PROTEIN ANNOTATION CORRECTNESS

Benjamin K. Nelson, MS

University of Nebraska, 2012

Advisor: Dhundy Bastola, Ph.D

Advances in sequencing technology have resulted in the sequencing of whole
genomes from many simple organisms such as fungi and bacteria, while allowing the
assembly of much more complex genomes like humans and chimpanzees. Consequently,
association of segments of newly sequenced genomes to specific function (i.e.
annotation) is being completed by comparative study of protein coding regions from
previously annotated genome data. While this is an ideal procedure to process and
annotate huge number of available genomic sequences, this approach can potentially lead
to propagating erroneous annotation in a public sequence repository and vastly diminish
the integrity of these new annotation of genome sequences. In this project, the WRongly
Annotated Protein identification System (WRAPS) has been created to analyze
previously annotated proteins quickly and efficiently. The likeliness that the protein is
correctly annotated is determined by weighted scoring schema based on conservation of
protein domain, the domains present in different reading frames, and isoelectric point. A study of 88,023 proteins of Yersinia, Staphylococcus, and Bacillus using WRAPS show that there are several proteins that can be considered wrongly annotated, as well as the correctness of annotation among these proteins.
ACKNOWLEDGEMENT

I would like to thank Dr. Dhundy Bastola for guiding me through the bioinformatics research process, so that I have an unshakable foundation within this multidisciplinary field. I would also like to thank the thesis advisory committee members Dr. Sanjukta Bhowmick and Dr. Zhengxin Chen for their input on my methods as well as their own ideas. I am also thankful for Mohammad Shafiullah, Ishwor Thapa and Dr. Joseph Steele for aiding in my hardware and software problems. I would also like to thank Sanjit Pandey for laying groundwork with the GARBASE project. Finally, I would like to thank the Bioinformatics Interest Group for their continued support throughout the research process.
Table of Contents

ABSTRACT ............................................................................................................................ ii
Copyright Statement ........................................................................................................ iv
ACKNOWLEDGEMENT ....................................................................................................... v
Table of Contents .............................................................................................................. vi
Tables ................................................................................................................................ viii
Figures ........................................................................................................................... viii
Equations ........................................................................................................................... viii
INTRODUCTION .................................................................................................................. 1
BACKGROUND ...................................................................................................................... 4
  DNA ................................................................................................................................. 4
  PROTEIN .......................................................................................................................... 6
  ALIGNMENTS .................................................................................................................... 10
  AUTOMATED ANNOTATION METHODS ........................................................................ 12
PUBLIC DATABASES ............................................................................................................ 14
  Conserved Domain Database ....................................................................................... 14
  GenBank ......................................................................................................................... 15
PROGRAMMING LANGUAGES .......................................................................................... 16
  Perl ................................................................................................................................. 16
  BioPerl ............................................................................................................................ 16
  MySQL ............................................................................................................................ 17
METHODS ............................................................................................................................ 17
  Overview ......................................................................................................................... 17
  Parallelization .................................................................................................................. 21
  Evidence 1 ....................................................................................................................... 21
  Evidence 2 ....................................................................................................................... 22
  Evidence 3 ....................................................................................................................... 23
  Evidence 4 ....................................................................................................................... 24
  Evidence 5 ....................................................................................................................... 26
  Evidence 6 ....................................................................................................................... 26
  Weighting ........................................................................................................................ 27
  Scoring ............................................................................................................................ 28
RESULTS AND DISCUSSION ................................................................. 29
References .......................................................................................... 38
APPENDIX A ....................................................................................... 43
APPENDIX B ....................................................................................... 45
APPENDIX C ....................................................................................... 50
Legend ................................................................................................. 50
Appendix D ......................................................................................... 52
Tables
Table 1 .................................................................................................................. 7
Table 2 .................................................................................................................. 33
Table 3 .................................................................................................................. 35
Table 4 .................................................................................................................. 36

Figures
Figure 1 ................................................................................................................... 1
Figure 2 ................................................................................................................... 5
Figure 3 ................................................................................................................... 8
Figure 4 ................................................................................................................... 18
Figure 5 ................................................................................................................... 22
Figure 6 ................................................................................................................... 23
Figure 7 ................................................................................................................... 24
Figure 8 ................................................................................................................... 25
Figure 9 ................................................................................................................... 27
Figure 10 ............................................................................................................... 37

Equations
Equation 1 ............................................................................................................. 28
INTRODUCTION

In science, the scientific method is used to solve problems. This method enables a scientist to test a property within the natural world. To do this, information is needed to first define a hypothesis and to test that hypothesis itself. In the fields of proteomics and genomics, this method holds true as well. In these sciences, one of the most common information are sequences, both DNA and protein. Since the discovery of DNA, many methods have been developed to sequence DNA and proteins for different organisms. Over time, these methods have become quicker and cheaper leading to an exponential increase in the number of sequences available. One of the leading databases, GenBank, showed an exponential rise in sequences being added to it, now doubling in size every 18 months, as seen in Figure 1 [1].

![Increase in GenBank Entries](image)

*Figure 1: Increase in the number of sequences added to GenBank*

While DNA and protein sequences are useful by themselves, often times an adjoining annotation is needed. An annotation tells useful things about the sequence. A
DNA sequence annotation may have information about coding sequences, the protein product of these coding sequences, and the protein translation of these coding sequences [2]. A protein sequence may have information about particular domains and molecules that bind to the protein. To discover the annotations, the traditional method is to experiment with these proteins within a biological lab [2]. While this method is very thorough, it is very slow. With the exponential increase in the number of sequencing projects, annotating these sequences using the traditional method is falling more and more behind.

To solve this problem automated methods were used to keep up with the increasing wave of sequence data being introduced. Systems such as GLIMMER and GeneMark help fill in the annotations for the abundance of sequences [3]. These methods create new annotations by comparing genes from the sequences to similar previously annotated genes already available in the database. This method is very fast and effective in annotating new sequences. However, it has been continuously introducing error to databases. If a gene is incorrectly annotated, then subsequent similar genes will also be incorrectly annotated. As incorrectly annotated genes are used in the annotation process, the error propagates. In fact, the percent of misannotated sequences of new sequences submitted to public databases has been rising [4]. This causes a problem with research as annotations of proteins must be validated as to avoid erroneous results.

To help reverse this problem by identifying the incorrect annotation, the Wrongly Annotated Protein Identification System (WRAPS) has been created. This system tests each protein on six different qualities that would be found in most proteins. Each of these evidence tests are weighted for each genus. The resulting weights are used to score
individual proteins. A protein's score will range from zero to one, giving the probability that a protein is correctly annotated. By using these scores, proteins may be flagged as incorrectly annotated so that flagged proteins may receive more stringent review and possibly be removed from the pool of possible proteins to be used in homology searches. Once the incorrectly annotated proteins are identified, the contents of public databases can be improved.

WRAPS has been created with certain goals in mind. First and foremost, this system is to provide an accurate measure into the correctness of a protein. Next, this system should be fast, as to avoid some of the pitfalls of its predecessor, GARBASE [5]. Next, this is to take advantage of biological properties of proteins and their corresponding nucleotides. Today, there are methods to identify wrongly annotated genomes. However, most are concerned only with computational aspects surrounding the methods used to annotate them [5, 6, 7]. While this may work better in a pipeline, these methods use the same set of tools resulting in the incorrect annotation in the first place. By using biological properties of proteins, this allows for new models to be made to further enrich protein annotation. Next, this system should be able to process information quickly, as annotation quality control means nothing if it falls victim to the flaws of traditional annotation in a wet lab. This is accomplished by parallelizing the evidence tests upon individual proteins. This greatly improves the amount of time it takes for WRAPS to run. This is because of the numbers of protein sequences to be tested within the database, which range from 50,000 to 300,000 in different jobs. Finally, this system should be able to be portable across many machines, provided that you have access to a MySQL database. This is accomplished by storing configurations in a file and accepting
configurations as command line arguments, so that the programs themselves do not have to be changed when using different computers.

**BACKGROUND**

**DNA**

Deoxyribonucleic acid, or better known as DNA, is the basic cellular information for living cells. DNA is comprised of two strands [9]. Each of these strands is a polymer, meaning it is a chain of repeated molecules. This polymer is structured as a series of nitrogenous bases attached to a sugar backbone. While the sugar backbone is a constantly repeated pattern, the bases vary. In fact, there are four different nitrogenous bases, namely adenine (A), cytosine (C), guanine (G), and thymine (T) [9]. Together, these bases are called nucleotides. By varying the sequence of these nucleotides, the DNA codes for different cellular products. Between the two strands, each of the nucleotides in one strand is matched up with the complement base in the opposite strand. For instance, the A nucleotide will match to the T nucleotide in the opposite strand and the C nucleotide will match to the G nucleotide in the opposite strand. The opposite of these pairings are also true between the two DNA strands. While both strands are comprised of the same molecules, they are oriented in opposite directions. One strand is called the 5’ to 3’ strand, while the other is called the 3’ to 5’ strand. This is because of the structure of the sugar backbone in DNA. The sugar takes the form of a ring, and each of the carbon atoms can be numbered from one to five. On the five carbon, there is a phosphate group that is attached to it. This phosphate is also attached to a three carbon on the next phosphate group. Depending on the way that this is read (from left to right),
the five carbon will be bonded to the phosphate, which is then bonded to the three carbon for a “5 to 3” strand, or the opposite direction for the “3 to 5 strand”. The chemical structure of a DNA strand can be seen in Figure 2. When studying DNA within bioinformatics, DNA is represented by a letter string of A, C, G, and T to designate the order of these nitrogenous bases in the 5’ to 3’ direction. It is not necessary to have the sequence of the 3 to 5 strand, as this can be derived by taking the complement of the 5 to 3 strand and reversing it.

Figure 2: Chemical structure of DNA [36]

Within a DNA sequence, there are important subsequences within it are called genes. Genes code for proteins. Proteins are polymers of discrete subunits called amino acids. Much like DNA, proteins are different, based on the amino acids used in the sequence. To be able to utilize the information within DNA, another molecule is made called ribonucleic acid (RNA) [10]. This serves as an intermediate to be taken to the ribosomes, which are proteins which facilitate the translation of DNA. RNA is much like DNA, only with a few differences. The first major difference is that RNA is only single
stranded, unlike DNA, which is double stranded. The second major difference is RNA has the uracil (U) nucleotide instead of the T nucleotide. The third major difference is RNA uses a different sugar molecule to use as the background.

PROTEIN

With each amino acid in a protein sequence, it has a corresponding sequence of three contiguous nucleotides. Each of these 3-mers is called a codon. Each of these 64 possible codons code for an amino acid [10,11,12]. Within humans, most bacteria, and many other organisms there are 20 different amino acids to be coded for [11,12]. Since not all of the possible codons can code for a unique amino acid, this means that most amino acids are coded for by multiple codons. For most amino acids, there are on average three different codons that can code for it. However, there are two amino acids, methionine and tryptophan, that have only one way of being coded. There are also three special codons called “stop” codons which terminate translation of the protein.

In a DNA sequence, a gene works only when the codons are in the correct frame. The frame of a DNA sequence defines which nucleotide 3-mers together form a codon. There are six different frames for a nucleotide, being 0, -0, 1, -1, 2, and -2. For instance, in the 0 frame, the codons begins with the first nucleotide in the sequence. In the 1 frame, the codons begin with the second nucleotide in the sequence. In the 2 frame, the codons begin with the third nucleotide in the sequence. In the negative frames, instead of dealing with the normal 5' to 3' strand, the complement 3' to 5' strand is used. When a different frame is used, called a frameshift, this can radically affect the outcome of the amino acids within the created protein, which can cause a non-functional protein. An example of all six frameshifts can be found below in Table 1. In this figure, adjoining
Table 1: Demonstration of all six frameshifts of a nucleotide sequence

<table>
<thead>
<tr>
<th>Frame</th>
<th>Nucleotide Sequence</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GCCTATGC</td>
<td>AY</td>
</tr>
<tr>
<td>2</td>
<td>CCTATGC</td>
<td>PM</td>
</tr>
<tr>
<td>3</td>
<td>CTATGC</td>
<td>LC</td>
</tr>
<tr>
<td>-1</td>
<td>GCATAGGC</td>
<td>A(STOP)</td>
</tr>
<tr>
<td>-2</td>
<td>CATAGGC</td>
<td>HR</td>
</tr>
<tr>
<td>-3</td>
<td>ATAGGC</td>
<td>IG</td>
</tr>
</tbody>
</table>

Within an organism, codons coding for the same amino acid will not always be used equally. In fact, in many organisms, some specific codons will be preferred over others when coding for a single amino acid. This is called codon bias. Codon bias happens due to transfer RNA (tRNA). This molecule helps construct the protein by adding an amino acid to the protein chain being assembled. There are 64 different tRNA's possible, each with the complementary sequence for a given codon. In a cell, it is more efficient to prefer one codon over another if they code for the same amino acid because it means that one tRNA can be produced more than others [12].

The proteins that are coded by genes are needed in performing activities within the cell. One such example is an enzyme. An enzyme has an active site that is specific to a molecule. Once the specific molecule is attached, an action is performed. For instance,
a molecule could have an active group cleaved. Another function a protein can provide is structure. For example, this could be hair, fur, or chitin. Proteins for a particular organism are most often not unique to that organism. In fact, across many species, a particular protein could exist. This is called conservation. Conservation does not only apply to entire proteins; it may also apply to amino acids within a protein, genes or even subsequences of the genome of an organism. A conserved protein or sequence means this is very beneficial to the organism's survival or perhaps even lethal in its absence. Within a genome, a special class of conservation exists. This is called synteny. Synteny is the conservation of gene order [35]. The order conserved can be forward or backwards, provided that the order is maintained. All possible ways that synteny can exist is shown below in Figure 3.

Figure 3: Demonstration of synteny. Protein sets 1, 2 and 3 exhibit synteny while protein set 4 does not exhibit synteny with any other protein set.
Within a protein, there are four different types of structure. The first is primary structure. This is only the sequence of amino acids of the protein. The secondary structure is the structures that the amino acids as a result of being connected to each other. There are two main secondary structures. They are alpha helix and beta sheet. As the names suggest, the alpha helix is a helical structure and the beta sheet is a flat sheet of amino acids. The tertiary structure is a three dimensional structure of the protein. This is due to event such as the protein burying its hydrophobic residues within itself or the cysteine amino acid forming a disulfide bond with another cysteine. Finally, the quaternary structure is how the subunits of the protein, called domains, fit together. Domains are a structure that is often repeated within multiple proteins that most often have a function associated with them, such as ion channels and active sites [13].

Domains are an important part of discovering what a protein does. In fact, protein function can usually be summarized by the domain(s) that the protein contains. Most often, these domains are conserved amongst homologous proteins. The function of a domain within a protein is crucial for the protein to be of use within the cell. Currently, there is a database called the Conserved Domain Database (CDD) that contains well studied domains [13]. If a domain is found within the CDD, it will have annotations explaining exactly what that domain is responsible for within a protein.

While domains are important within a protein, the individual amino acids carry a certain weight towards its characteristics. One example of this is the charges provided by individual amino acids. These are either positively charged or negatively charged. These contribute to the total electric charge of the protein. When a protein is in a solution, the charge is affected by the pH of the solution. The point in which the charge of the protein
is neutral is called the isoelectric point. This property is important because proteins with
certain isoelectric points will be localized to certain areas within the cell [14]. This idea
can be expanded to protein function. Proteins with similar function will be localized to a
similar location. Because of this, the proteins will have a similar isoelectric point.
Inversely, if two proteins have the same isoelectric point, they may be located in the cell
in similar locations.

ALIGNMENTS

In bioinformatics, sequences can be compared to each other using an alignment.
An alignment is a way to compare protein or DNA sequences. An alignment is
accomplished by laying two or more sequences side by side so that the best match
between them is accomplished. By adding gaps and shifting parts of the sequences, a
“best fit” is reached called the optimal alignment. An alignment enables the
bioinformatist to show structural or functional similarities between the sequences [15].
There are two main ways of aligning sequences, which are global and local alignments.
By using a global alignment, the two sequences are compared as a whole [16]. To create
the optimal alignment, gaps are introduced into either sequence. A good example of a
global alignment algorithm is the Needleman-Wunsch algorithm[16].

A local alignment differs from a global alignment by focusing on subsequences
within the sequences being aligned that are identical or very similar [17]. It is from these
subsequences that match that the alignment is built around. From here, gaps are
introduced into both sequences to allow an optimal alignment. A good example of a local
alignment algorithm is the Smith-Waterman algorithm [37]. It is actually from this local
algorithm that the famous BLAST algorithm [17] is based.
BLAST is most likely one of the most influential algorithms within bioinformatics. BLAST was developed by National Center for Biotechnology Information (NCBI) as a way to find sequences that align closely with a query sequence [17]. BLAST can be used with both nucleotide and protein sequences. This is done by dividing the query sequence into words of a certain length (k-mers). For DNA sequences, this length is usually 11 nucleotides. Regions within sequences that are repetitions of the same elements are ignored. From here, the database of sequences to be searched is examined for these same words. Once a sequence with an identical word is found, this word is extended in both sequences, at both sides of each subsequence. These words are extended as long as the aligned words are above a certain predefined cutoff score. From here, these matching words or high-scoring pairs, as the BLAST documentation call them, from the same sequence are combined to create alignments. For each alignment created, a number called the E value is given. The E value [38] is a measure of how close the aligned sequences are, by showing the number of times an unrelated sequence in the database would be aligned in this way by chance. The lower E value is for the hit sequence, the better the alignment will be between the two sequences.

BLAST itself is actually just the basic algorithm in a series of programs provided by NCBI. Amongst them are BLASTN and BLASTP which work with nucleotides and proteins respectively. There are also related programs such as PSI-BLAST and RPS-BLAST. In the case of PSI-BLAST, this uses the results of a BLASTP search against a database to create a matrix that shows the conservation between the alignments [18]. This can find divergent evolutionary relationships. In the case of RPS-BLAST, a number of matrices like those generated in a PSI-BLAST, are queried against a reference
sequence [19]. This allows the user to find conserved evolutionary motifs within the sequence, such as domains.

AUTOMATED ANNOTATION METHODS

In the world today, there are many different pipelines that are used in the automatic annotation of putative proteins from prokaryotic organisms [7, 20]. Despite the differences in algorithms, these pipelines there are two major steps involved in protein annotation.

The first step in the annotation process is gene finding, or structural annotation [7, 20]. Most gene finding algorithms make use of Markov models to help predict genes. A Markov model is a series of states, with the first $n$ states determining the probability of the next given state. For a Markov model of $n$ order, the previous $n$ states in the sequence are used to predict the next state. Within a DNA sequence, each of the nucleotides is a state. An example could be a 5th order Markov model with the current sequence of nucleotides being ATGTA. The probability that the the next nucleotide will be a A or a G is most likely going to be much lower than the probability that the next nucleotide will be a C or a T. To use a Markov model, it must be trained by using a set of data in which it can discover the probability of each base after every possible set of $n$ bases. For the Markov model to work, it needs to have $4^{n+1}$ probabilities calculated. As the order gets higher, this is much more demanding, as each of the sets of $n +1$ bases are required. Due to this problem, one gene finder has dominated the field. This gene finder is GLIMMER [7]. GLIMMER uses an Interpolated Markov model (IMM), rather than a normal Markov model. IMMs hold an advantage over normal Markov models due to the number of states required to be calculated a given probability of the next state. For instance, if
the current sequence contains AGT, and the probability calculated from a 3\textsuperscript{rd} order model that the next state will be G is the same probability as if the current sequence contains GT as a 2\textsuperscript{nd} order model, there is no benefit to training the model to the 3\textsuperscript{rd} order over the 2\textsuperscript{nd}. This enables GLIMMER to only have to use larger order models on individual rare cases [20].

The second step in the annotation process is functional annotation [7,21]. Several tools are available here to find an annotation previously defined that fits the unannotated protein like BLAST, BLAST Extended Repraze (BER), InterPro, and hidden Markov models. Of these methods, the simplest is BLAST. In this context, a simple BLASTX search from the putative gene's nucleotide sequence or a BLASTP search from the putative gene's protein sequence against a protein database would potentially find homologous proteins and therefore give a functional annotation for the putative gene in question. The next method, BER helps correct some of the errors within the putative gene, such as frameshift mutations and premature stop codons as well as extending the nucleotide sequence by adding additional bases on both of the upstream and downstream ends of the gene [7]. This can be especially helpful in cases where the organism in which you sequenced the DNA expressed a rare phenotype caused by a mutation like this. To use a HMM to find functional annotations, first HMMs are trained from known members that share a specific function. Next the putative gene is scored from the HMM. If this gene score highly enough within the HMM, it can be considered a member of the family. The next tool that can be used to discover functional annotation is a database such as InterPro. These databases house multiple sequence alignments on a single feature [22]. This feature may be a domain, site, or a shared functional family.
Each of these steps is critical in the current execution of an automatic annotation pipeline. Currently there are many different automated annotation pipelines in production. Among them are the IGS annotation engine [7] and the J. Craig Venter Institute (JCVI) annotation service [6]. To predict a correct protein annotation using the IGS pipeline, GLIMMER and RNA prediction are used to find the structural annotations while BLAST variants, BER, and HMMs are used to find functional annotations. The data gathered in each step is then used to determine whether the protein is hypothetical or has enough supporting data to be considered a true protein. When the JCVI pipeline is used to predict protein annotations, GLIMMER identifies putative genes within a genomic sequence that are thought to encode proteins. Next, BLAST, BER, HMMs are used to compare these with existing families of proteins. If a putative protein fits a family profile, it is considered to be part of the family.

PUBLIC DATABASES
Conserved Domain Database
The Conserved Domain Database (CDD), as the name suggests, is a database containing domains conserved and repeated across many different proteins. This database has been collected and manually curated by the National Center for Biotechnology Information (NCBI) to ensure complete accuracy [13]. Using Entrez, this database is linked to other NCBI databases. These domains are three dimensional structures within proteins that can help determine the protein’s function within the cell. The CDD first began as a mirror of other popular domain databases, such as Pfam. Pfam is a domain databases that contain profile Hidden Markov Models for different domain families built from multiple sequence alignments [23]. The CDD has since differentiated
itself from databases like Pfam by using domain hierarchies. For instance, a protein may match many domain Hidden Markov Models from Pfam, but only a small number of domain Hidden Markov Models from CDD [13]. This is because the profile Hidden Markov Models in Pfam can be based on sequences aligned on many properties. For instance, this could be a common structural domain between proteins that have very different functions. To avoid this pitfall, the CDD uses a hierarchical grouping of protein domains. The protein in question is compared to a tree of domain profile Hidden Markov Models, so that the search starts out broad, but ends up on a specific set of domains. The CDD is available for download onto local systems via an FTP site. The database can be created on a local machine by using the “formatrpsdb” command within the NCBI Software Development Toolkit.

GenBank

GenBank is a public database curated by NCBI [24]. This database holds protein and DNA sequences submitted by both large and small laboratories, as well as their annotations. Sequence can be obtained as proteins, nucleotides, or even whole genomes. GenBank is accessible by the Entrez, which allows users to search for sequences as well as do BLAST searches. All of the data that is available online is also available to be downloaded locally via an FTP site. The files within GenBank are in GenBank format, as seen in Appendix A. Files in GenBank format contain biological annotations as well as bibliographical annotations surrounding the sequence(s) in question. As of February 2012, there were 1,498,192,468 total sequences in GenBank [1].
PROGRAMMING LANGUAGES

Perl
Perl is an interpreted programming language developed by Larry Wall in the late 1980's [25]. The original purpose of this programming language was to process reports, making this language efficient at processing text. Perl became popular language to use due to its ease of use, partially due to the lack of type checking used in other scripting languages [26]. By having most all primitive data types interchangeable as a single type, the scalar, this saves developer’s time. In the bioinformatics realm, this programming language has widespread use, most likely due to the ease of learning, the available code modules on the Internet, and most importantly, the ability to easily deal with text documents. The ability to deal with text is especially important within bioinformatics because this is the format of most all publicly available data.

BioPerl
While Perl is very helpful within bioinformatics, this results in many different users doing the same thing, while all writing different code to do this. To solve this problem, a collection of Perl modules called BioPerl have been created to help within the bioinformatics domain [27]. To accomplish this, BioPerl uses an object oriented approach. For example, in the SeqIO module, an annotated sequence can be read from a file into an object in memory. From memory, methods can be called to view the properties of a sequence. BioPerl solves many bioinformatics problems, such as parsing files, indexing databases, and using other bioinformatics tools like BLAST. By having modules such as BioPerl, this saves developers a great deal of time, as code does not need
to be written a second time.

MySQL
MySQL is an open source relational database management system [34]. It uses
the SQL query language to perform database transactions.

METHODS
Overview
The main goal of WRAPS is to input annotated proteins specified by the user and
to output a score that indicates how likely the protein for which it codes is properly
annotated. To accomplish this goal, this system is divided into four discrete sections. A
simplified outline of this system can be found in Figure 4. A more in-depth diagram of
this system can be found in Appendix B, as well as diagrams for each of the scripts that
help set up and execute WRAPS.
Figure 4: Structure of WRAPS

The first step in WRAPS is database population. This step involves placing information for each protein sequence into several tables within the MySQL database. An entity relationship diagram for the database can be found in Appendix C. To accomplish this, two Perl scripts are organized in a master-slave format, so that database population can be easily parallelized. The first script handles user input. The user inputs database credentials, the name of the database to be populated, and the directories in which the sequence data, in GenBank format [39], is located. An example of a file in the GenBank format can be found in Appendix D. This script recursively descends into the directories to find all possible files that have valid GenBank extensions (.seq, .gbk). It
then reads a configuration file to find the host location and the maximum number of allowed child processes to use. For each file in the input directories, a child process is spawned which runs a second script through use of the fork system call. The second script uses BioPerl to parse the GenBank file and extract data pertaining to each coding sequence within the genome. This extracted data is inserted into the specified MySQL database. Once the maximum number of slave processes has been created, the master process waits until one of the slave processes has died. Once a slave has died, the master will create another slave process to run the second script upon the next file. This process continues until all of the valid GenBank files have been processed by the second script.

The second section of WRAPS tests each protein upon the evidence criteria. This step involves reading information from the MySQL database protein entries, performing tests upon the data, and inserting a value indicating the outcome of each evidence test on each protein. This is implemented in a series of Perl scripts. This section is also structured in a master-slave format. The first Perl script, the launcher, reads user information, being the database credentials and the database to be used. When the launcher script runs, it reads from the WRAPS configuration file the number of slave processes to use and the number of proteins that have been populated within the selected database. For each protein, the launcher executes a fork system call and executes an instance of a second Perl script, the evidence caller. A number of slave processes are created by the launcher and limited to the maximum number of slave processes specified in the configuration file. When a slave process dies, a new slave process is spawned to run the evidence caller script upon the next protein until all of the proteins have been run with the evidence caller script. Within the evidence caller script, a series of scripts from a
designated directory is called on the specific protein in question. Each of the called scripts atomically runs a single evidence test upon the protein and returns “PASS” if the protein passed the evidence test and “FAIL” if the protein failed the evidence test. The individual evidence tests will be discussed more thoroughly later in this paper. After each evidence test is completed, the result, a MySQL script is created to record the outcome of the evidence test upon the protein the script was run. This is not immediately inserted in database as any evidence test results do not affect any other evidence test results, so the results of each evidence test can be recorded into the database after the completion of all evidence tests upon each protein. By using this practice, the I/O load on the disk is considerably lessened, which allows the parallel evidence testing processes to analyze proteins faster.

For third section of WRAPS the evidence weights associated with each family of related sequence are calculated. This is done by a Perl script. Each group of related sequences is grouped by genus. For instance, *Yersinia pestis* and *Yersinia pseudotuberculosis* will both be grouped in the Yersinia family. Using the number of proteins that have passed the evidence tests within the positive control the weights will be calculated. The actual equation used to calculate the weights will be discussed later in this paper.

In final section of WRAPS, the scores for each protein are calculated. This is done by a pair of Perl scripts. This section is structured in a master-slave format, with the first Perl script using the fork system call to assign a unique protein to the second Perl script within each slave process. The method used to calculate the score will be discussed later in this paper. After WRAPS has been run, the resulting scores for each
protein can be found in the MySQL database.

**Parallelization**

To parallelize WRAPS, a simple method involving the "fork" system call was used. By running the WRAPS launcher script a pool of child processes are spawned in which each child process handles the evidence test evaluations for a single protein. This enables WRAPS to run much faster than only evaluating evidence tests in series. Since the output of one evidence test affects no other concurrently running evidence tests, this effectively compartmentalizes individual proteins. This is advantageous as the failure of the evaluation of the evidence tests on a single protein does not affect the evaluation of the evidence tests upon any other protein. Due to the atomic nature of the evidences tests used upon each protein, this allows for very simple implementation. In the launcher script, the parent process keeps spawning child processes to evaluate proteins. The parent will keep track of the number of children it has while only spawning up to a user defined limit. Once that limit is reached, the parent will wait until a child has died, and then spawn the next process. This process is repeated until there are no additional proteins to be tested. After this, the parent waits for its child processes to die before terminating as to avoid "zombie" processes.

Evidence 1

The first evidence is to search the protein against the Conserved Domain Database (CDD) by using rps-blast. The structure of this evidence can be found in Figure 3. For a protein, its three dimensional structure is made up of smaller functional units called domains [13,28]. The CDD is a database that contains domains that are evolutionarily important to different organisms that are considered correct [13]. The CDD is manually
curated, so it does not have the same failings as databases like GenBank. When a protein has a domain found within the CDD, this is an indication that this protein is correct, so it passes this evidence. When there are no hits for a protein, this is an indication that the protein is incorrectly annotated. The structure of this evidence can be found in Figure 5.

![Diagram of Evidence 1](image)

*Figure 5: Structure of Evidence 1*

Evidence 2

The second evidence analyzes the protein’s corresponding genomic sequence. A correct protein should have conserved domains within its current frame. If a protein in a different frame contains a conserved domain, this indicates that the alternate frame is most likely the correct frame, meaning that the current annotation is incorrect. For this evidence each of the protein's nucleotides are translated in six different frameshifts being 0, 1, 2, -0, -1, and -2. An example each of the six possible frameshifts, with the codons in alternating colors, can be seen in Table 1. To avoid missing pieces of the alternate frames’ proteins, the flanking intergenic genomic sequences are added to the protein’s
corresponding genomic sequence. If there is a domain found within any of the frameshifts, this is counted as failing the evidence. If there are no domains found within the frameshifts, then this protein passes the evidence. The structure of this evidence can be seen in Figure 6.

Evidence 3
The third evidence examines the synteny of protein, which provides the relative position of the protein in the organism genome and it expected to be conserved among similar organisms [29]. To evaluate the synteny, the strandedness and the order of protein will be evaluated. To determine if a protein exhibits similar synteny with its flanking protein sequences against a homologous protein with similar flanking sequences, each of the flanking sequences of the protein in question will be used in a BLAST search against a BLAST database made of proteins from the same genus as the proteins being tested.
For each hit from the protein on the 5' end of the protein being tested, it is matched up with each other protein hit from the protein on the 3' end. If any of these combinations of two proteins are flanking one protein, this protein's synteny is conserved and it passes this evidence. If it had failed, then the protein fails the evidence. The structure of this evidence can be found in Figure 7.

![Diagram of evidence structure](image)

**Figure 7: Structure of Evidence 3**

**Evidence 4**

The fourth evidence views the codon bias of the protein. Recall that a codon is a set of three sequential bases from the previous section. Since there are four total bases and three bases comprise a codon, it is easy to see that there are 64 possible codons. However, there are only 21 amino acids for which can be coded for. This means that there are multiple codons which code for the same amino acid or stop codon. There are 22 different amino acids that can be produced from a given codon, which means each amino
acid or stop codon has an average of 3 ways that it can be coded. This means a protein of 100 amino acids could be coded by about $5 \times 10^{47}$ different nucleotide sequences [30]. However, not all these codons will be evenly used to code for the same protein, as discussed before with codon bias. This is an evolutionary response to different forces such as GC content or limiting metabolic load due to tRNA synthesis [30,31]. By having a codon bias, an organism can produce more of the same tRNA to code for the same amino acid, resulting in less energy needed to differentiate tRNA molecule. Between similar organisms, a similar codon bias is exhibited [30]. To show this similarity, an algorithm will be developed that clusters like proteins based on codon bias into a binary tree. The evidence 4 will be collected based on whether the protein will cluster with organism from the same genus. If a protein is clustered with its genus, then it passes the evidence, otherwise it will be considered to fail this evidence. The structure of this evidence can be found in Figure 8.

![Figure 8: Structure of Evidence 4](image-url)
Evidence 5

The fifth evidence analyzes the isoelectric point of the protein in comparison to the expected normal confines of isoelectric point. This is found by using the Henderson-Hasselbach equation [32]. To determine the isoelectric point of the protein an algorithm as described by Tabb having $O(\log(n))$ time [32] will be implemented. Since most all proteins are expected to have an isoelectric point between 4 and 10, protein showing the isoelectric point outside of the range is expected to be wrongly annotated [33].

Evidence 6

The final evidence analyzes the isoelectric point of the protein in comparison to other proteins from organisms of the same genus. Within a cell, proteins that have a similar location and function will have a similar isoelectric point [33]. By using a BLAST search, a set of homologous proteins are found within the genus. If any of those proteins matches the isoelectric point of the protein within a certain degree of error, then this protein passes the evidence, otherwise, the protein fails the evidence. The structure of this evidence can be found in Figure 9.
Protein i

Find homologous proteins (BLASTP)

Do homologous proteins exhibit a similar isoelectric point?

Yes

Pass

No

Fail

Figure 9: Structure of Evidence 6

Weighting

Finally, the overall score for any annotation will be decided based on a weighting system. The purpose of the weighting system is to give more emphasis to evidence that apply more strongly to a families. For instance, a family may have an abnormally high isoelectric point average, but most all of the proteins within that family have domains that are recorded within the CDD. In this case, the weight applied to Evidence 1 would be weighted more heavily, by receiving a larger multiplier and Evidence 5 would be weighted less heavily, by receiving a smaller multiplier. To do this, each of the protein descriptions are checked to find a set of proteins that are known to be correctly annotated, known as the positive control. If a protein has been given an Enzyme Commission number, this protein is considered correctly annotated and used within the positive
control. The set of proteins belonging to the positive control are determined in the
database population step. For each genus, the frequencies of all passing evidence tests
are summed together. To give the weight for each evidence test, the frequency of the
number of proteins from the positive control passing evidence test is divided by the sum
of all the frequencies for all evidences. This weight shows the importance of a given
evidence property within a given genus. The equation for determining the evidence
weights for a family can be found in Equation 1.

\[
W_i = \frac{Pc_i}{\sum_{i=1}^{n} Pc_i}
\]

Equation 1a

\[
S = \sum_{i=1}^{n} W_i b_i
\]

Equation 1b

\[\text{Pc}_i - \text{Positive control protein } i\]

\[W_i - \text{Weight for evidence test } i\]

\[b_i - \text{Binary coefficient for evidence } i\]

Scoring

To determine the score of an individual protein, the respective weight is
multiplied by a coefficient based on whether the protein passed or failed the given
evidence. The coefficient will be 0 if the protein failed and 1 if the protein passed.
Finally, all of these products will be summed together to determine the protein's score.
From this, one can determine how closely this protein is to being correctly annotated. By
giving a score as opposed to a binary number, this can show which sequences need to be
re-annotated there by increasing the reliability of sequence annotation. The equation for
determining the score can be found in Equation 2.
Statistical Analysis

To examine each of the evidence tests and the data as a whole, a series of statistical tests were performed to determine the data’s statistical significance. For each evidence test, a Chi-squared test [40] was performed, comparing the proportion of the proteins in the positive control passing the evidence test to the proportion of proteins not included in the positive control passing the evidence test. The Chi-squared test compares an expected value to the observed value. This test was chosen to examine the difference between the positive control set and the case study set. Proportional values were used rather than the true frequencies, so that the large numbers did not skew the Chi-squared tests. This Chi-squared test is to measure the representation of passed evidence tests in the positive control sample versus the representation in the case study sample. For each set, the proportion of the positive control proteins passing the evidence (13356 samples tested) is compared against the proportion of the case study proteins passing the evidence test. If the Chi-squared test shows a statistical significance between the positive control sample and the case study sample, this means that the quality examined within the evidence test is important for the validity of the protein. If the Chi-squared test is shown to be not statistically significant, then we must conclude that there is no significant difference in the pass rate for that particular evidence test between the positive control and the case study, showing that this evidence is a weak identifier for protein validity.

In addition to the Chi-squared test, a two-sampled, two tailed Z-test [41] was used to compare the distributions of scores. This test enables the comparison of each set of protein scores to each other, based upon the standard deviations. This test was chosen to compare the ranges of scores of the positive control and the case study sets. Rather than
only comparing a single facet of the score, this test enables us to see the entire ranges of scores. If the Z-test shows a statistical significance between the two sets of scores, and the z-value is positive, this shows that there is a clear difference between the positive control samples and case study samples, meaning that there is an increased level of correct proteins in the case study samples compared to the positive control samples. If the Z-test shows a statistical significance between the two sets of scores, and the z-value is negative, this shows that there is a clear difference between the positive control samples and case study samples, meaning that there is a decreased level of correct proteins in the case study samples compared to the positive control samples. If the Z-test shows no statistical significance between the two sets of scores, and the z-value is negative, this shows that there is no clear difference between the positive control samples and case study samples.

**RESULTS AND DISCUSSION**

WRAPS was used to analyze 88,023 proteins belonging to the Yersinia, Staphylococcus, and Bacillus genera available on University of Nebraska at Omaha's KLAB server, which partially serves as a database for all bacterial genomes found in January 2011 version of GenBank. For this set, 13356 proteins have been manually annotated, both structurally and functionally. In Appendix A, a table of the results of each possible outcome from the evidence tests can be found. For this set, the positive control set was used as a template for which to predict the whole set. From this, the amount in each category was expected to grow in proportion equal to the magnitude of the increase in size of the test set. We would expect this because of the protein homology that exists within closely related prokaryotes. In the prokaryotes used in the study, such
as *Yersinia pestis*, these organisms have many proteins existing within the positive control. In closely related organisms, we would expect that homologous proteins would be found within the genomes. These homologous proteins would be expected to exhibit the same properties as their cousins within the positive control. However, viewing the difference in sizes between the expected column and the observed column, it is clear that the data set is not as correctly annotated as the positive control set.

To help with the understanding of this data, I submit two terms. The first term to be defined is evidence class. This is row in the table, representing a single possible outcome on all of the evidence tests, which can be expressed by a bit string. For instance, score class could be passing the first evidence test, passing the second evidence test, failing the third evidence test, passing the fourth evidence test, passing the fifth evidence test and failing the sixth evidence test, which would result in a bit string of 110110. In total, there are 64 possible evidence classes, as there are two discrete possible outcomes of each evidence test. The second term to be used within the context of this data is annotational correctness. This is represented by the score given. A higher score shows that an evidence class shows a higher annotational correctness. It should also be noted that the table is ordered by the evidence classes, not the annotational correctness. It is important to note that proteins within the positive control may belong to evidence classes which have a lower level of annotational correctness than others. In the case of the positive control, this is the data in which the weighting system is trained. Even amongst correctly annotated proteins there will be variance. However, the variance between annotational evidence classes in the whole data set will vary much more, as we would image that proteins are missannotated within the set.
In this data set, it should be noted that the proportions that the different evidence class are zero-sum, meaning, for every decrease in a single evidence class's proportion, one or more other evidence classes must increase in proportion within the data set. In the system presented, the addition of unverified proteins drops the frequency of evidence classes corresponding to more correct annotations, with an increase in the frequency of evidence classes corresponding to less correct annotations. To introduce a change in evidence class frequency, this would be caused by the additional passing or failing from the evidences. For each of the evidence class frequencies that decrease in comparison to the positive control, this is due to the failure of more evidence tests, as noted by the general decrease in the annotational correctness of the entire data set when compared to the positive control. For the frequency of a higher annotational correctness evidence class to drop and the frequency of a similar evidence class of lesser annotational correctness, it is more probable that a single test was failed rather than the multiple tests.

When examining evidence classes, such as 111111 and 1111110, it can be seen that there is a decrease in proportion of proteins at this evidence class when compared to the positive control. It should be noted that when comparing these two bit strings to 111101 and 111100 respectively, the evidence classes that are different from the first two evidence classes by failing evidence 5. This can also be observed for many of the other evidence classes exhibiting a high annotational correctness. For the evidence tests needed to fail to change the frequencies of evidence classes, evidence 1 and 5 were failed the most. This mirrors the weights assigned to the individual evidences, as the weights assigned to evidence 1 and 5 are the highest of all evidences. This shows that these properties tested by the evidence 1 and 5 give a protein a higher degree of annotational
correctness than other properties, and are properties exhibited in correctly annotated proteins. Table 2, which is a table of the weights calculated along with the sensitivity and the number of proteins from each of the evidence test passing from both the full data set and the positive control. For the tests, it was not possible to calculate the specificity of the tests, as there are no known negatives without testing in a laboratory, since known wrongly annotated proteins would not be knowingly inserted into GenBank. In a side experiment, randomly generated DNA sequences were used to synthesize proteins to be tested. However, in each of the 1000 sequences generated, each failed every evidence test but evidence 2.

<table>
<thead>
<tr>
<th></th>
<th>Bacillus</th>
<th>Staphylococcus</th>
<th>Yersinia</th>
<th>Positive Control</th>
<th>Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ev1</td>
<td>0.301</td>
<td>0.295</td>
<td>0.246</td>
<td>13332</td>
<td>0.998203</td>
</tr>
<tr>
<td>ev2</td>
<td>0.071</td>
<td>0.055</td>
<td>0.046</td>
<td>2563</td>
<td>0.191899</td>
</tr>
<tr>
<td>ev3</td>
<td>0.036</td>
<td>0.013</td>
<td>0.102</td>
<td>3775</td>
<td>0.282645</td>
</tr>
<tr>
<td>ev4</td>
<td>0.142</td>
<td>0.259</td>
<td>0.202</td>
<td>10784</td>
<td>0.807427</td>
</tr>
<tr>
<td>ev5</td>
<td>0.299</td>
<td>0.288</td>
<td>0.240</td>
<td>13017</td>
<td>0.974618</td>
</tr>
<tr>
<td>ev6</td>
<td>0.150</td>
<td>0.088</td>
<td>0.162</td>
<td>7208</td>
<td>0.539683</td>
</tr>
</tbody>
</table>

Based on the evidence presented, the most important properties that were known domains as shown by evidence 1, having an isoelectric that fits a typical range as shown by evidence 5, and by having a codon bias that is also expressed by another member of the same genus. In the case of evidence 1, it was quite obvious how important having a well-known domain, as in the evidence class table, as there are hardly any proteins that did not contain a domain not known to the CDD. In the case of evidence 5, it was not nearly as pronounced as evidence 1, but when viewing the evidence class table, there is a
significant drop in the frequency of evidence class members in the positive control whenever evidence 5 is failed when compared to the evidence classes that did pass evidence 5. While this test is a good indication of a protein’s annotation’s correctness, I believe that it could be refined so that it more reflects the individual organisms, as providing just a flat range of isoelectric may incorrectly fail the proteins from more extremophilic prokaryotes. In the case of evidence 4, it is shown much in the same way as evidence 5, but not nearly as pronounced. This evidence can also be improved with adding more flexibility to the bias model, by allowing one or two mismatches between the protein's bias bit string and matching bit strings. As a tradeoff, a certain number of matching strings would need to be able to be matched for the protein to pass this evidence test.

Despite the successes with evidence tests 1, 4, and 5, evidence tests 2, 3, and 6 admittedly need more work. To begin, evidence test 3 is probably the hardest test of all of the evidence tests to be able to prove without, as to increase its lenience, would greatly increase its complexity. One such way would be to allow small numbers of proteins within the proteins exhibiting synteny. In the case of evidence 6, this evidence test suffers from the unproven functional annotations. To improve this, rather than finding a homologous protein from all of the proteins within the database, only homologous proteins from the positive control should be used. This would ensure that homologous proteins that are found truly have a homologous function. The final sub-par evidence as shown by the execution of the system is evidence 2. Unlike its sister evidence, evidence 1, it is a negative evidence, so rather than looking for the presence of the domains, it is looking for the absence in the alternate frames. To improve this evidence, the genomic
location of domains can be noted and queried against the database. In the case that proteins would overlap, this could be checked so that finding true protein within another true protein would not constitute a failure.

<table>
<thead>
<tr>
<th>Evidence Test</th>
<th>Positive Control vs All Proteins</th>
<th>Positive Control vs Case study</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( P = 4.90 \times 10^{-5} )</td>
<td>( P = 1.71 \times 10^{-6} )</td>
</tr>
<tr>
<td>2</td>
<td>( P = 0.83 )</td>
<td>( P = 0.80 )</td>
</tr>
<tr>
<td>3</td>
<td>( P = 0.92 )</td>
<td>( P = 0.90 )</td>
</tr>
<tr>
<td>4</td>
<td>( P = 0.88 )</td>
<td>( P = 0.85 )</td>
</tr>
<tr>
<td>5</td>
<td>( P = 0.49 )</td>
<td>( P = 0.41 )</td>
</tr>
<tr>
<td>6</td>
<td>( P = 0.84 )</td>
<td>( P = 0.81 )</td>
</tr>
</tbody>
</table>

Examining the Chi-squared tests (Table 3), the difference in correctness in the positive control samples and the case study samples was shown to be statistically significant in evidence test 1. In the other evidence tests, the difference positive control proteins and the case study proteins was shown to be not statistically significant. The results from these Chi-squared tests, show us that there is a significant difference only in the conserved domains located in the positive control proteins and the case study proteins suggesting that the case study samples are enriched with incorrect annotations. Since we are almost certain that the case study set contains incorrectly annotated proteins, we must conclude that this is the reason for the decreased proportion in the proteins passing this evidence. This confirms to us that having a domain that is known to be correct is a defining feature within correctly annotated proteins.

In the other evidence tests, this shows that there is no significant difference between the case study proteins and the positive control proteins. While these results may seem to show that evidence tests 2, 3, 4, 5, and 6 do not matter between sets of
correctly annotated and incorrectly annotated proteins, it is important to note that we do
know the true annotations of the case study proteins. It is extremely likely that many of
these proteins are correctly annotated. If we use any evidence test on any two sets of
correctly annotated proteins, we would expect that there should be no statistically
significant difference between the result of either evidence tests. In the sample sets used,
the ratio of incorrectly annotated proteins to total sample set can easily change the
statistical significance of the evidence tests. In addition to the makeup of the case study
set, the statistical insignificance of the case study set may also be due to the tuning of the
individual evidence tests. For instance, a minority of both the positive control and the
case study passed evidence test 3. This lack of passed tests is also reflected within the
sensitivity shown in Table 2. This suggests that this evidence test is too stringent. To
correct this, it could be changed to allow a number of other genes within the genes
exhibiting synteny (within a threshold), as long as the three proteins exhibiting synteny
maintain their order.

<table>
<thead>
<tr>
<th>Table 4: Z-test of protein scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>z-value</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>Case study vs Positive Control</td>
</tr>
<tr>
<td>All proteins vs Positive Control</td>
</tr>
</tbody>
</table>

Examining the results of the Z-test (Table 4), shows that the difference between
the case study proteins and the positive control is statistically significant. A histogram
showing the amount of proteins relative each of the 64 possible scores can be seen below.
in Figure 10. In this histogram, it is plain to see that there is a reduction in the scores when comparing the case study set to the positive control set. This reduction in scores and a statistically significant difference suggest that there are wrongly annotated proteins within the case study data set. When adding this to the analysis done with the Chi-squared test, this suggests that while only one evidence test was statistically significant, the aggregation of all of the evidence tests show that there is a larger difference between the positive control data set and the case study data set.

Figure 10: Histogram of scores from positive control and case study proteins

In pipelines currently being used in sequencing today, the data generated from WRAPS would greatly improve the quality of the annotated sequences that are released.
While I do not believe WRAPS will be adopted by many, if any pipelines, the tests adapted within it can be used within current protein annotation pipelines. By using an automated method that quickly and efficiently identifies possible missannotated proteins to pipelines, this brings a dramatic increase in speed compared to traditional wet-lab protein annotation, and more accuracy to the current methods of automated annotation.

References
171:737-738

10. Volkin E and Astrachan L, “Phosphorus incorporation in *Escherichia coli* ribonucleic acid after infection with bacteriophage T2”, Virology(1956), 2:149-161


31. Sharp P and Li W, “The codon adaptation index - a measure of direcitonal


36. “DNA chemical structure”.


40. Pearson, Karl. "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be
reasonably supposed to have arisen from random sampling”. Philosophical Magazine Series 5 (1900) 302:157–175

# APPENDIX A

<table>
<thead>
<tr>
<th>ev1</th>
<th>ev2</th>
<th>ev3</th>
<th>ev4</th>
<th>ev5</th>
<th>ev6</th>
<th>Yersinia</th>
<th>Staphylococcus</th>
<th>Bacillus</th>
<th>Avg. Score</th>
<th>Pos. Control</th>
<th>Expect</th>
<th>Observed</th>
<th>Obs. Growth/Expect</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>7</td>
<td>46</td>
<td>496</td>
<td>10.75</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.15</td>
<td>0.30</td>
<td>0.29</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>222</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.30</td>
<td>0.29</td>
<td>0.22</td>
<td>0.27</td>
<td>1</td>
<td>7</td>
<td>943</td>
<td>143.08</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.45</td>
<td>0.59</td>
<td>0.51</td>
<td>0.51</td>
<td>0</td>
<td>0</td>
<td>498</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.14</td>
<td>0.03</td>
<td>0.08</td>
<td>0.08</td>
<td>2</td>
<td>13</td>
<td>280</td>
<td>21.24</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.29</td>
<td>0.33</td>
<td>0.37</td>
<td>0.33</td>
<td>0</td>
<td>0</td>
<td>370</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.44</td>
<td>0.31</td>
<td>0.31</td>
<td>0.35</td>
<td>4</td>
<td>26</td>
<td>936</td>
<td>35.51</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.59</td>
<td>0.61</td>
<td>0.59</td>
<td>0.60</td>
<td>1</td>
<td>7</td>
<td>1106</td>
<td>167.82</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.04</td>
<td>0.06</td>
<td>0.26</td>
<td>0.12</td>
<td>0</td>
<td>0</td>
<td>34</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.19</td>
<td>0.36</td>
<td>0.55</td>
<td>0.37</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.33</td>
<td>0.34</td>
<td>0.49</td>
<td>0.39</td>
<td>0</td>
<td>0</td>
<td>74</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.49</td>
<td>0.65</td>
<td>0.77</td>
<td>0.63</td>
<td>0</td>
<td>0</td>
<td>23</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.18</td>
<td>0.08</td>
<td>0.35</td>
<td>0.20</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.33</td>
<td>0.39</td>
<td>0.64</td>
<td>0.45</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.48</td>
<td>0.37</td>
<td>0.57</td>
<td>0.47</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.63</td>
<td>0.67</td>
<td>0.86</td>
<td>0.72</td>
<td>0</td>
<td>0</td>
<td>130</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.07</td>
<td>0.30</td>
<td>0.07</td>
<td>0.15</td>
<td>0</td>
<td>0</td>
<td>473</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.22</td>
<td>0.60</td>
<td>0.35</td>
<td>0.39</td>
<td>0</td>
<td>0</td>
<td>258</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.37</td>
<td>0.59</td>
<td>0.29</td>
<td>0.41</td>
<td>3</td>
<td>20</td>
<td>2081</td>
<td>105.25</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.52</td>
<td>0.89</td>
<td>0.57</td>
<td>0.66</td>
<td>0</td>
<td>0</td>
<td>891</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.21</td>
<td>0.33</td>
<td>0.15</td>
<td>0.23</td>
<td>0</td>
<td>0</td>
<td>533</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.36</td>
<td>0.63</td>
<td>0.44</td>
<td>0.48</td>
<td>0</td>
<td>0</td>
<td>434</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.51</td>
<td>0.61</td>
<td>0.37</td>
<td>0.50</td>
<td>0</td>
<td>0</td>
<td>1882</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.66</td>
<td>0.91</td>
<td>0.66</td>
<td>0.75</td>
<td>2</td>
<td>13</td>
<td>1632</td>
<td>123.81</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.11</td>
<td>0.36</td>
<td>0.33</td>
<td>0.27</td>
<td>0</td>
<td>0</td>
<td>88</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.26</td>
<td>0.66</td>
<td>0.62</td>
<td>0.51</td>
<td>0</td>
<td>0</td>
<td>55</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.41</td>
<td>0.65</td>
<td>0.55</td>
<td>0.53</td>
<td>2</td>
<td>13</td>
<td>286</td>
<td>21.70</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.56</td>
<td>0.95</td>
<td>0.84</td>
<td>0.78</td>
<td>0</td>
<td>0</td>
<td>207</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0.39</td>
<td>0.42</td>
<td>0.35</td>
<td>0</td>
<td>0</td>
<td>26</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.40</td>
<td>0.69</td>
<td>0.70</td>
<td>0.60</td>
<td>0</td>
<td>0</td>
<td>155</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.55</td>
<td>0.67</td>
<td>0.64</td>
<td>0.62</td>
<td>0</td>
<td>0</td>
<td>213</td>
<td>NA</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.70</td>
<td>0.97</td>
<td>0.92</td>
<td>0.87</td>
<td>2</td>
<td>13</td>
<td>741</td>
<td>56.22</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.30</td>
<td>0.03</td>
<td>0.08</td>
<td>0.13</td>
<td>17</td>
<td>112</td>
<td>559</td>
<td>NA</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.45</td>
<td>0.33</td>
<td>0.36</td>
<td>0.38</td>
<td>13</td>
<td>86</td>
<td>446</td>
<td>5.21</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.60</td>
<td>0.31</td>
<td>0.30</td>
<td>0.40</td>
<td>868</td>
<td>5721</td>
<td>6007</td>
<td>1.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.75</td>
<td>0.61</td>
<td>0.58</td>
<td>0.65</td>
<td>684</td>
<td>4508</td>
<td>2922</td>
<td>0.65</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.44</td>
<td>0.05</td>
<td>0.16</td>
<td>0.22</td>
<td>87</td>
<td>573</td>
<td>1965</td>
<td>3.43</td>
</tr>
</tbody>
</table>
APPENDIX B

Setting Configurations

WRAPS Settings

WRAPPconfiguration.pl

WRAPS.config

User credentials

WRAPcreate.pl

Creating Database tables

MySQL DB

Adds tables
User credentials

Start Protein

MySQL DB

WRAPlauncher.pl

Protein id

WRAPS_start.pl

ev1.pl

MySQL DB

wraps.sql

Evaluation of properties

Pass/fail

ev2.pl

... 

evn.pl

MySQL DB
Evidence 3

MySQL DB

Get Proteins

Protein 1 fasta

Ev3.pl

Protein 2 fasta

blastp

Genus BlastDB

Results 1

Find proteins that have p_ids differing by 2

Results 2

Evidence 4

MySQL DB

ev4.pl

Find synonymous codon usage (same bit string)
Evidence 5

MySQL DB
Get Protein
ev5.pl
Test isoelectric range

Evidence 6

MySQL DB
Get Protein
ev6.pl

Genus BlastDB
blastp
Results
Homologous proteins
Compare isoelectric point
APPENDIX C

Legend

**genome.g_id** – Identification number assigned to an individual GenBank file by the genome.database
**genome.gacc** – Genomic accession number from GenBank
**genome.genus** – Genus of the organism
**genome.path** – path to GenBank file
**genome.ts** – timestamp

**protein.p_id** - Identification number assigned to an individual protein
**protein.g_id** – foreign key referencing genome.g_id, corresponds to the genome in which the protein belongs
**protein.pacc** – protein accession number
**protein.p_desc** – description of the protein
**protein.start** – nucleotide start position of the protein
**protein.end** – nucleotide end position of the protein
**protein.strand_id** – number assigned to sequential proteins on the same strand. 3’ to 5’
strand is negative while the 5’ to 3’ strand is positive
protein.pseq – protein sequence
genweight.genus – genus for weighs
genweight.wn – weight for evidence test n

positive.p_id – protein id for positive control protein

bias.p_id – protein id for protein
bias.bit_string – bit string representing preferred codon usage
bias.bit_num – number of codons in which the protein holds a bias for

hom.p_id – protein id for protein
hom.hom_id – protein id for homologous protein

trans.p_id – protein id for protein
trans.nseq – nucleotide sequence for protein

intergenic.p_id – protein id for protein
intergenic.p_id5 – 5’ neighboring protein (if any)
intergenic.p_id3 – 3’ neighboring protein (if any)
intergenic.inter5 – 5’ intergenic sequence
intergenic.inter3 – 3’ intergenic sequence

evn.p_id – protein id for protein
evn.pass – Boolean variable representing protein passing (1) or failing (0) evidence n
## Appendix D

<table>
<thead>
<tr>
<th>LOCUS</th>
<th>SCU49845</th>
<th>5028 bp</th>
<th>DNA</th>
<th>PLN</th>
<th>21-JUN-1999</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFINITION</td>
<td>Saccharomyces cerevisiae TCP1-beta gene, partial cds, and Axl2p (AXL2) and Rev7p (REV7) genes, complete cds.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACCESSION</td>
<td>U49845</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VERSION</td>
<td>U49845.1</td>
<td>GI:1293613</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KEYWORDS</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOURCE</td>
<td>Saccharomyces cerevisiae (baker's yeast)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORGANISM</td>
<td>Saccharomyces cerevisiae Eukaryota; Fungi; Ascomycota; Saccharomycotina; Saccharomycetes; Saccharomycotales; Saccharomycetales; Saccharomyces.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REFERENCE 1</td>
<td>(bases 1 to 5028)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTHORS</td>
<td>Torpey,L.E., Gibbs,P.E., Nelson,J. and Lawrence,C.W.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TITLE</td>
<td>Cloning and sequence of REV7, a gene whose function is required for DNA damage-induced mutagenesis in Saccharomyces cerevisiae</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOURNAL</td>
<td>Yeast 10 (11), 1503-1509 (1994)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PUBMED</td>
<td>7871890</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REFERENCE 2</td>
<td>(bases 1 to 5028)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTHORS</td>
<td>Roemer,T., Madden,K., Chang,J. and Snyder,M.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TITLE</td>
<td>Selection of axial growth sites in yeast requires Axl2p, a novel plasma membrane glycoprotein</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOURNAL</td>
<td>Genes Dev. 10 (7), 777-793 (1996)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PUBMED</td>
<td>8846915</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REFERENCE 3</td>
<td>(bases 1 to 5028)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTHORS</td>
<td>Roemer,T.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TITLE</td>
<td>Direct Submission</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOURNAL</td>
<td>Submitted (22-FEB-1996) Terry Roemer, Biology, Yale University, New Haven, CT, USA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEATURES</td>
<td>Location/Qualifiers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>source</td>
<td>1..5028</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/organism</td>
<td>&quot;Saccharomyces cerevisiae&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/db_xref</td>
<td>&quot;taxon:4932&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/map</td>
<td>&quot;IX&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/chromosome</td>
<td>&quot;IX&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDS</td>
<td>&lt;1..206</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/product</td>
<td>&quot;TCP1-beta&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/protein_id</td>
<td>&quot;AAA98665.1&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/db_xref</td>
<td>&quot;GI:1293614&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/translation</td>
<td>&quot;SSYNGISTSGLDLNNGTIADMRQLGIVESYKLKRAVVSSASEA AEVLLRVDNIIRARPRTANQHM&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gene</td>
<td>687..3158</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDS</td>
<td>687..3158</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/gene</td>
<td>&quot;AXL2&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/codon_start</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/product</td>
<td>&quot;AXL2p&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/protein_id</td>
<td>&quot;AAA98666.1&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/db_xref</td>
<td>&quot;GI:1293615&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| /translation| "MTQLQISLLLTLATLSSLHHLVWATPYEAYPGKQYPFWAVRNWESF TPQISNDYKSSVDTQATAITYNCDFDLPSWLSDDSSRTFGEPSPSDDLSDJNTLLFY VILEGTDADLTSNLNTYQFVPVTNRPSLSSDPINLALLNKYTGNYAALKNLPMF VNFNTFDRSMFTNEISISYXGRSYLNPWLFDESEKFGTATPVTINSAIAP ESYFVIIADIEGFSAVEVEFELVIGAHQILRTTSQSLIINVDTGNYSDLPLNMLV YLDDDPISSDGLSINLNDAPDWADWATISGSPVFELLGLKSNPANFSFSSIWYDYG DVYIFNEFFVTDDLFPISSLINATGREGFWFSYFLPSQFTDYVNTNVLNFLEFFNSOQ DHDVQFFQSNLTLAGEVPKNDKLGLKANQGQSQELYFNNIAGMSKITHSSHASA NATSTRSHHTSTSSYTSSTYTAKISSTSAAATSSAPAALPAANKTSSHNNKAVAIA ..
CGVAIPLGVILVALICFLIFWRRRRENPDDENLPHAISGPDLNNPANKPNQENATPLNPFDDDASSYDTSTAI
RRLAALNTLKLHATSDEISSVDEKRDSLGMNTYNDQQ
SQQSKELLAKPPFQEEPSFDQPQNRSSSSVYMDSEPVAKNWSRVTQDSVDDSDS
QGSKKTVDFTEKLDLEAFEKEKRTSDVTMSSLDFWNSNSPSVRSKSFPSNYVT
HRRNLQMQDSQSKNGNITFTNTSSTSSDFVPKDGENCFWHSMFEDFRSRFSK
LDFSNKSNVNVGQVYDKHRIPEML"